Oracle 9i RMAN 参考使用手册

作者概述：

陈吉平，网络 ID 为 piner，现在任职于淘宝网(www.taobao.com)，主要负责网站后台数据库维护，擅长备份与恢复，数据库高可用性与容灾，对数据库优化也有很深的研究。希望能广交 oracle 朋友，共同进步。

概述

RMAN的概念与体系结构
Oracle 9i RMAN的配置
运用RMAN进行备份
运用RMAN进行还原与恢复
RMAN的日常管理
RMAN的恢复目录

一、RMAN 的概念与体系结构

Recovery Manager（RMAN）是一种用于备份(backup)、还原(restore)和恢复(recover)数据库的 Oracle 工具。RMAN只能用于ORACLE8或更高的版本中。它能够备份整个数据库或数据库部件，如表空间、数据文件、控制文件、归档文件以及Spfile参数文件。RMAN也允许您进行增量数据块级别的备份，增量RMAN备份是时间和空间有效的，因为他们只备份自上次备份以来有变化的那些数据块。而且，通过RMAN提供的接口，第三方的备份与恢复软件如veritas将提供更强大的备份与恢复的管理功能。

通过RMAN，也提供了其它更多功能，如数据库的克隆、采用RMAN建立备用数据库、利用RMAN备份与移动裸设备（RAW）上的文件等工作将变得更方便简单。9i的RMAN通过增强的自动配置与管理功能，以及特有的块级别的恢复，将使备份与恢复工作变得更加快捷与完美。9i的RMAN有如下特征特性：

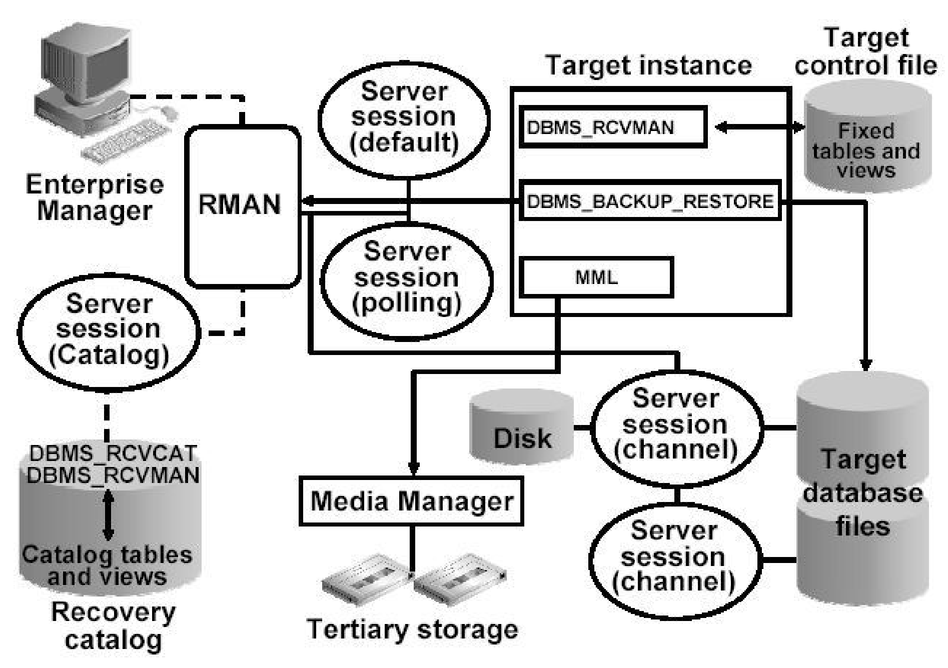
- 自动的备份与恢复
- 方便的备份归档日志
- 自动检测新的数据文件
- 支持增量备份
- 最大限度的减少备份与恢复的错误
- 减少恢复的时间
- 在热备份中不会产生额外的redo日志
- 腐烂数据块的自动检测

• 并行的备份与恢复操作

• 在线备份时，表空间不用置于备份模式

可以看到，在以上的一些特性中，显示了RMAN强大的功能与好处，以上功能的实现，是因为RMAN是块级别的备份与恢复，备份与恢复发生在数据库块级别，可以通过比较数据块而获得一致性的数据块，可以避免备份没有用过的块，可以检验块是否腐烂等块级别的问题。

RMAN的体系结构可以用如下图表示：



RMAN 的体系结构

对于组成以上 RMAN 的结构，说明如下：

1、RMAN 工具

也就是 RMAN 命令，起源于 Oracle 版本 8，一般位于$ORACLE_HOME/bin 目录下，可以通过运行 rman 这个命令来启动 RMAN 工具，用于备份与恢复的接口。

2、服务进程

RMAN 的服务进程是一个后台进程，用于与 RMAN 工具与数据库之间的通信，也用于 RMAN 工具与磁盘/磁带等 I/O 设置之间的通信，服务进程负责备份与恢复的所有工作，在如下情况将产生一个服务进程

• 当连接到目标数据库

• 分配一个新的通道

3、通道

通道是服务进程与 I/O 设备之前读写的途径，一个通道将对应一个服务进程，在分配通道时，需要考虑 I/O 设备的类型，I/O 并发处理的能力，I/O 设备能创建的文件的大小，数据库文件最大的读速率，最大的打开文件数目等因素

4、目标数据库

就是 RMAN 进行备份与恢复的数据库，RMAN 可以备份除了联机日志，pfile，密码文件之外的数据文件，控制文件，归档日志，spfile

5、恢复目录

用来保存备份与恢复信息的一个数据库，不建议创建在目标数据库上，利用恢复目录可以同时管理多个目标数据库，存储更多的备份信息，可以存储备份脚本。如果不采用恢复目录，可以采用控制文件来代替恢复目录，oracle 9i 因为控制文件自动备份的功能，利用控制文件很大程度上可以取代恢复目录。

6、媒体管理层

Media Management Layer (MML)是第三方工具或软件，用于管理对磁带的读写与文件的跟踪管理。如果你想直接通过 RMAN 备份到磁带上，就必须配置媒体管理层，媒体管理层的工具如备份软件可以调用 RMAN 来进行备份与恢复。

7、备份，备份集与备份片

当发出 backup 命令的时候，RMAN 将创建一个完成的备份，包含一个到多个备份集，备份集是一个逻辑结构，包含一组的物理文件。这些物理文件就是对应的备份片。备份片是最基本的物理结构，可以产生在磁盘或者磁带上，可以包含目标数据库的数据文件，控制文件，归档日志与 spfile 文件。

备份集与备份片有如下规定

一个数据文件不能跨越一个备份集，但是能跨越备份片

数据文件，控制文件能保存在同样的备份集上，但是不能与归档日志保存在同样的备份集上

二、启动与运行 RMAN

2.1 运行要求

1、进程与内存要求

更多的进程的需要

大池的分配

2、基本环境变量需求

ORACLE_SID, ORACLE_HOME, PATH, NLS_LANG, 如果用到了基于时间的备份与恢复，需要另外设置 NLS_DATE_FORMAT

3、权限要求

需要 SYSDBA 系统权限

如果是本地，可以采用 OS 认证，远程需要采用密码文件认证

4、版本要求

RMAN 工具版本与目标数据库必须是同一个版本，如果使用了恢复目录，还需要注意

- 创建 RMAN 恢复目录的脚本版本必须等于或大于恢复目录所在数据库的版本
- 创建 RMAN 恢复目录的脚本版本必须等于或大于目标数据库的版本

2.2 基本运行方法

9i 默认是 nocatalog，不使用恢复目录，使用命令 rman 即可进入 RMAN 的命令行界面，如

[oracle@db oracle]$ $ORACLE_HOME/bin/rman

Recovery Manager: Release 9.2.0.4.0 - Production
Copyright (c) 1995, 2002, Oracle Corporation. All rights reserved.
RMAN>
连接目标数据库，可以用如下类似命令
RMAN>Connect target /

2.3 怎么样运行 RMAN 命令

1、单个执行
RMAN>backup database;
2、运行一个命令块
RMAN> run {
2> copy datafile 10 to
3> '/oracle/prod/backup/prod_10.dbf';
4> }
3、运行脚本
$ rman TARGET / @backup_db.rman
RMAN> @backup_db.rman
RMAN> RUN { @backup_db.rman }
运行存储在恢复目录中的脚本
RMAN> RUN { EXECUTE SCRIPT *backup_whole_db* };
4、SHELL 脚本，如果在 cron 中执行，注意设置正确的环境变量在脚本中

```
[oracle@db worksh]$ more rmanback.sh
#!/bin/ksh
#set env
export ORACLE_HOME=/opt/oracle/product/9.2
export ORACLE_SID=test
export NLS_LANG="AMERICAN_AMERICA.zhs16gbk"
export PATH=$PATH:$ORACLE_HOME/bin

echo "----------------------------start----------------------------";date
#backup start
$ORACLE_HOME/bin/rman <<EOF
connect target
delete noprompt obsolete;
backup database format '/netappdata1/rmanback/tbdb2/%U_%s.bak' filesperset = 2;

exit;
EOF
echo "------------------------------end------------------------------";date
```

三、RMAN 的自动配置

Oracle 9i 可以配置一些参数如通道，备份保持策略等信息，通过一次设定可以多次使用，而且，设置中的信息不影响脚本中的重新设置。RMAN 默认的配置参数，通过 show all 就可以看出来。

RMAN> show all;
RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1;
CONFIGURE BACKUP OPTIMIZATION OFF;
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE CONTROLFILE AUTOBACKUP OFF;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR TYPE DISK TO '%F';
CONFIGURE DEVICE TYPE DISK PARALLELISM 1;
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1;
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1;
CONFIGURE MAXSETSIZE TO UNLIMITED;
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/u01/app/oracle/product/9.0.2/dbs/snapcf_U02.f';

3.1 备份策略保持

分为两个保持策略，一个是时间策略，决定至少有一个备份能恢复到指定的日期，一个冗余策略，规定至少有几个冗余的备份。

CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 5 DAYS;
CONFIGURE RETENTION POLICY TO REDUNDANCY 5;
CONFIGURE RETENTION POLICY TO NONE;

在第一个策略中，是保证至少有一个备份能恢复到 Sysdate-5 的时间点上，之前的备份将标记为 Obsolete。第二个策略中说明至少需要有三个冗余的备份存在，如果多余三个备份以上的备份将标记为冗余。NONE 可以把使备份保持策略失效，Clear 将恢复默认的保持策略。

3.2 通道配置与自动通道分配

通过 CONFIGURE 配置自动分配的通道，而且可以通过数字来指定不同的通道分配情况。

CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/U01/ORACLE/BACKUP/%U‘
CONFIGURE CHANNEL n DEVICE TYPE DISK FORMAT '/U01/ORACLE/BACKUP/%U‘

当然，也可以在运行块中，手工指定通道分配，这样的话，将取代默认的通道分配。

```
Run{
allocate channel cq type disk format='/u01/backup/%u.bak'
……
}
```

以下是通道的一些特性
读的速率限制
Allocate channel …… rate = integer
最大备份片大小限制
Allocate channel …… maxpiecesize = integer
最大并发打开文件数（默认 16）
Allocate channel …… maxopenfile = integer

3.3 控制文件自动备份

从9i开始，可以配置控制文件的自动备份，但是这个设置在备用数据库上是失效的。通过如下的命令，可以设置控制文件的自动备份
CONFIGURE CONTROLFILE AUTOBACKUP ON;
对于没有恢复目录的备份策略来说，这个特性是特别有效的，控制文件的自动备份发生在任何 backup 或者 copy 命令之后，或者任何数据库的结构改变之后。
可以用如下的配置指定控制文件的备份路径与格式
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR TYPE DISK TO '%F';
在备份期间，将产生一个控制文件的快照，用于控制文件的读一致性，这个快照可以通过如下配置
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/9.0.2/dbs/snapcf_U02.f';

3.4 设置并行备份

RMAN支持并行备份与恢复，也可以在配置中指定默认的并行程度。如
CONFIGURE DEVICE TYPE DISK PARALLELISM 4;
指定在以后的备份与恢复中，将采用并行度为4，同时开启4个通道进行备份与恢复，当然也可以在run的运行块中指定通道来决定备份与恢复的并行程度。
并行的数目决定了开启通道的个数。如果指定了通道配置，将采用指定的通道，如果没有指定通道，将采用默认通道配置。

3.5 配置默认 IO 设备类型

IO 设备类型可以是磁盘或者磁带，在默认的情况下是磁盘，可以通过如下的命令进行重新配置。
CONFIGURE DEFAULT DEVICE TYPE TO DISK;
CONFIGURE DEFAULT DEVICE TYPE TO SBT;
注意，如果换了一种 IO 设备，相应的配置也需要做修改，如
RMAN> CONFIGURE DEVICE TYPE SBT PARALLELISM 2;

3.6 配置多个备份的拷贝数目

如果觉得单个备份集不放心，可以设置多个备份集的拷贝，如
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 2;
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 2;
如果指定了多个拷贝，可以在通道配置或者备份配置中指定多个拷贝地点
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/u01/backup/%U', '/u02/backup/%U';
RMAN>backup datafile n format '/u01/backup/%U', '/u02/backup/%U';

3.7 备份优化

可以在配置中设置备份的优化，如
CONFIGURE BACKUP OPTIMIZATION ON;
如果优化设置打开，将对备份的数据文件、归档日志或备份集运行一个优化算法。
同样的 DBID,检查点 SCN,ResetlogSCN 与时间
正常离线，只读或正常关闭的文件
归档日志
同样的线程，序列号 RESETLOG SCN 与时间

3.8 备份文件的格式

备份文件可以自定义各种各样的格式，如下
%c 备份片的拷贝数
%d 数据库名称
%D 位于该月中的第几天 **(DD)**
%M 位于该年中的第几月 **(MM)**
%F 一个基于 **DBID** 唯一的名称**,**这个格式的形式为 **c-IIIIIIIIII-YYYYMMDD-QQ,** 其中 **IIIIIIIIII** 为该数据库的 **DBID**，**YYYYMMDD** 为日期，**QQ** 是一个 **1-256** 的序列
%n 数据库名称，向右填补到最大八个字符
%u 一个八个字符的名称代表备份集与创建时间
%p 该备份集中的备份片号，从 **1** 开始到创建的文件数
%U 一个唯一的文件名，代表**%u_%p_%c**
%s 备份集的号
%t 备份集时间戳
%T 年月日格式**(YYYYMMDD)**

四、用 RMAN 备份

RMAN可以用来备份主或备用数据库，如表空间、数据文件、归档日志、控制文件、服务器文件与备份集。

4.1 文件拷贝

原始文件的拷贝，有点类似于 OS 热备份，可以拷贝整个数据文件到另外一个地点，但是结果仅仅只能写入到硬盘，而且单独的文件是分开的。

一个文件拷贝的例子

```
run {
allocate channel d1 type disk;
allocate channel d2 type disk;
allocate channel d3 type disk;
copy # first
datafile 1 to '$HOME/prd1.dbf',
datafile 2 to '$HOME/prd2.dbf';
copy # second
datafile 3 to '$HOME/prd3.dbf';
sql 'alter system archive log current';
}
```

4.2 备份与备份集

RMAN 的常规备份是产生只有 RMAN 才能识别的备份集，所以，除了 copy 命令之外的其他备份，都是 RMAN 产生的备份集以及对应的备份片。

一个备份数据库的例子，开启两个通道，将数据库备份到磁带

```
run {
allocate channel t1 type 'SBT_TAPE';
allocate channel t2 type 'SBT_TAPE';
backup
filesperset 2
format 'df_%t_%s_%p'
database;
}
```

RMAN 也可以实现多个镜相的备份

```
Run{
allocate channel d1 type disk;
allocate channel d2 type disk;
allocate channel d3 type disk;
SET BACKUP COPIES 3;
BACKUP DATAFILE 7 FORMAT '/tmp/%U','?/oradata/%U','?/%U';
};
```

以下是常见的备份归档的例子

```
RMAN>sql ‘alter system archive log current’;
RMAN>backup archivelog all delete input;
RMAN> backup archivelog from time '01-jan-00' until time '30-jun-00';
RMAN> backup archivelog like 'oracle/arc/dest/log%';
```

RMAN> backup archivelog all;
RMAN> backup archivelog from logseq 20 until logseq 50 thread 1;
RMAN> backup archivelog from scn 1 until scn 9999;

在 RAC 环境中，因为数据库是共享的，所以可以连接到一个实例就可以备份整个数据库，但是，因为归档日志可以备份在本地，所以 RAC 归档日志的备份就变的复杂一些，我们可以通过连接到两个实例的通道来备份两个实例的归档日志。

```
run{
ALLOCATE CHANNEL node_c1 DEVICE TYPE DISK CONNECT 'sys/pass@dbin1';
ALLOCATE CHANNEL node_c2 DEVICE TYPE DISK CONNECT 'sys/pass@dbin2';
sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
backup archivelog all delete input format '/u01/dbbak/%U_%s.bak' filesperset = 5;
}
```

4.3 常见备份参数

1、Keep 参数可以长期的保持特殊的备份或者拷贝，让它们不受默认备份保持策略的影响，如

RMAN> BACKUP DATABASE KEEP UNTIL TIME
2> "to_date('31-MAR-2002','DD_MM_YYYY)" nologs;
RMAN> BACKUP TABLESPACE SAMPLE KEEP FOREVER NOLOGS;

其中 NOLOGS 表示可以不保留该备份以来的归档日志，默认是 LOGS，表示保留该备份以来的参数，如果想让该备份永久有效，可以使用 FOREVER 参数。

2、Tag 参数指明了备份集的标志，可以达到 30 个字符长度，如

RMAN> BACKUP DEVICE TYPE DISK DATAFILE 1 TAG
2> "wkly_bkup";

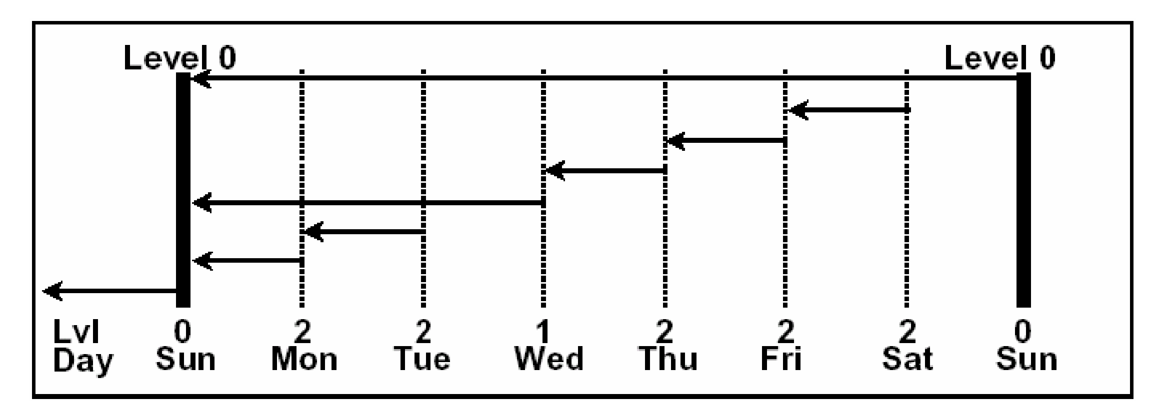
在 Oracle 92 版本以后，RMAN 自动提供一个 TAG，格式为 TAGYYYYMMDDTHHMMSS 如 TAG20020208T133437，通过备份标志 TAG，也可以很方便的从备份集进行恢复，如

Restore database from tag=’tag name’

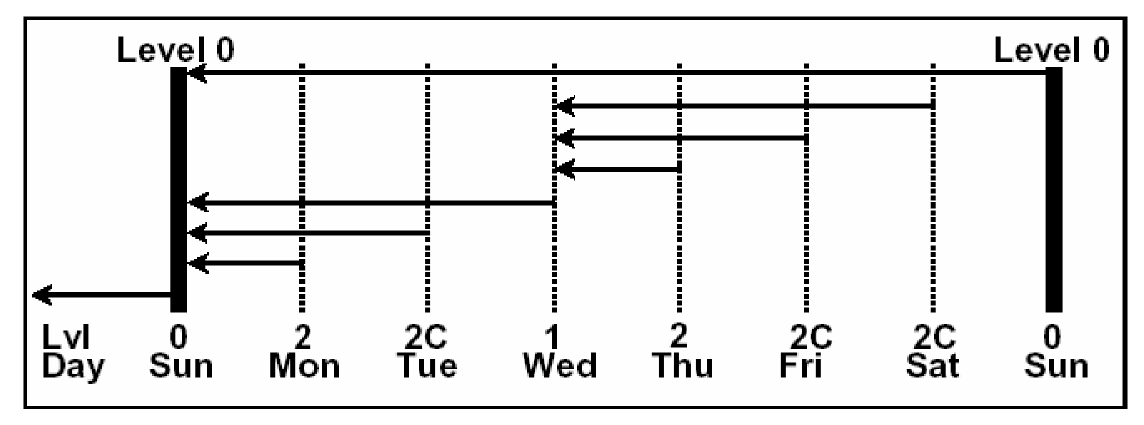
4.4 增量备份

在说明增量备份之前，首先要理解差异增量与累计增量备份，以及增量备份的备份与恢复原理。

差异增量，是默认的增量备份方式。



累计增量备份



可以看到，差异增量是备份上级或同级备份以来变化的块，累计增量是备份上级备份以来变化的块。累计增量增加了备份的时间，但是因为恢复的时候，需要从更少的备份集中恢复数据，所以，为了减少恢复的时候，累计增量备份将比差异增量备份更有效。

不管怎么样增量备份，在 Oracle 版本 9 中，还是需要比较数据库中全部的数据块，这个过程其实也是一个漫长的过程，而且由于增量备份形成多个不同的备份集，使得恢复变的更加不可靠而且速度慢，所以增量备份在版本 9 中仍然是鸡肋，除非是很大型的数据仓库系统，没有必要选择增量备份。

Oracle 版本 10 在增量备份上做了很大的改进，可以使增量备份变成真正意义的增量，因为通过特有的增量日志，使得 RMAN 没有必要去比较数据库的每一个数据块，当然，代价就是日志的 IO 与磁盘空间付出，完全还是不适合 OLTP 系统。另外，版本 10 通过备份的合并，使增量备份的结果可以合并在一起，而完全的减少了恢复时间。

增量备份都需要一个基础，比如 0 级备份就是所有增量的基础备份，0 级备份与全备份的不同就是 0 级备份可以作为其它增量备份的基础备份而全备份是不可以的，是否选择增量备份作为你的备份策略，最终，需要你自己有一个清醒的认识。

以下是零级备份的例子

backup incremental level 0 database;

一级差异增量例子

backup incremental level 1 database;

一级累计增量例子

backup incremental level 1 cumulative database;

4.5 备份检查

我们可以通过 Validate 命令来检查是否能备份，如数据文件是否存在，是否存在坏块不能被备份，如：

```
BACKUP VALIDATE DATABASE;
BACKUP VALIDATE DATABASE ARCHIVELOG ALL;
```

4.6 重新启动备份

对于异常结束了的备份，很多人可能不想再重新开始备份了吧，特别是备份到 90%以上，因为异常原因终止了该备份，那怎么办呢？RMAN 提供一个重新开始备份的方法，通过简单的命令，你就可以只备份那不到 10%的数据了。

```
RMAN> BACKUP NOT BACKED UP SINCE TIME 'SYSDATE-14'
2> DATABASE PLUS ARCHIVELOG;
```

4.7 RMAN 动态性能视图

以下是与 RMAN 备份有关系的一些动态性能视图，信息是从控制文件中获取的。

V$ARCHIVED_LOG
V$BACKUP_CORRUPTION
V$COPY_CORRUPTION
V$BACKUP_DATAFILE
V$BACKUP_REDOLOG
V$BACKUP_SET
V$BACKUP_PIECE
V$BACKUP_DEVICE
V$CONTROLFILE_RECORD_SECTION

这里还有一个视图，可以大致的监控到 RMAN 备份进行的程度。如通过如下的 SQL 脚本，将获得备份的进度。

```
SQL> SELECT SID, SERIAL#, CONTEXT, SOFAR, TOTALWORK,
2 ROUND(SOFAR/TOTALWORK*100,2) "%_COMPLETE"
3 FROM V$SESSION_LONGOPS
4 WHERE OPNAME LIKE 'RMAN%'
5 AND OPNAME NOT LIKE '%aggregate%'
6 AND TOTALWORK != 0
7 AND SOFAR <> TOTALWORK;
SID SERIAL# CONTEXT SOFAR    TOTAL    WORK %_COMPLETE
---    -------       -------         -------      ---------    ----------
13    75            1             9470       15360      61.65
12    81            1             15871      28160      56.36
```

五、还原与恢复

5.1 常规还原与恢复

RMAN 的整个恢复过程可以分为还原（restore）与恢复（recover），他们在含义上是有很大差别的，一个是指物理意义的文件的还原与拷贝，一个是指数据库一致性的恢复，所以，正确的理解这两个概念，有助于正确的恢复数据库。

对于 RMAN 的备份，还原操作只能是在用 RMAN 或 RMAN 包来做了，对于恢复操作则是很灵活的了，除了 RMAN，也可以在 SQLPLUS 中完成。还原与恢复一个数据库，可以用如下两个简单的命令完成

```
RMAN>restore database;
RMAN>recover database;
```

恢复一个表空间，或者恢复一个数据文件，相对比较恢复数据库可能花费更少的时间。

```
RMAN> SQL "ALTER TABLESPACE tools OFFLINE IMMEDIATE";
RMAN> RESTORE TABLESPACE tools;
RMAN> RECOVER TABLESPACE tools;
RMAN> SQL "ALTER TABLESPACE tools ONLINE";
```

对于数据库与数据文件，可以从指定的tag恢复

```
RMAN>RESTORE DATAFILE 1 FROM TAG=’tag name’
```

对于时间点恢复等不完全恢复，可能只有完全的还原数据库了。

```
RMAN> RUN {
2> ALLOCATE CHANNEL c1 TYPE DISK;
3> ALLOCATE CHANNEL c2 TYPE DISK;
4> SET UNTIL TIME = '2002-12-09:11:44:00';
5> RESTORE DATABASE;
6> RECOVER DATABASE;
7> ALTER DATABASE OPEN RESETLOGS; }
```

不完全恢复在 RMAN 中还可以用基于日志的恢复

```
RMAN> RUN {
2> SET UNTIL SEQUENCE 120 THREAD 1;
3> ALTER DATABASE MOUNT;
4> RESTORE DATABASE;
5> RECOVER DATABASE; # recovers through log 119
6> ALTER DATABASE OPEN RESESTLOGS;
7> }
```

如果有可能，也可以恢复数据文件到一个新的位置

```
SET NEWNAME FOR datafile
'/u01/oradata/tools01.dbf' TO '/tmp/tools01.dbf';
RESTORE datafile '/u01/oradata/tools01.dbf';
SWITCH DATAFILE ALL;
```

除了恢复数据库与数据文件，我们也可以恢复控制文件，需要启动到 nomount 下，用如下的命令即可

Restore controlfile from ‘file name’
Restore controlfile from autobackup
Restore controlfile from tag=‘……’
在正常情况下，不用恢复归档日志，恢复进程会自动寻找所需要的归档日志，当然我们也可以指定恢复到哪里。
SET ARCHIVELOG DESTINATION TO '/u02/tmp_restore';
RESTORE ARCHIVELOG ALL;
如果使用的服务器参数文件（spfile），RMAN 可以备份该参数文件，如果发生文件损坏，可以用 RMAN 恢复 spfile 参数文件，在没有参数文件的情况下，用 Rman 的临时参数文件启动数据库到 Nomount 下，执行如下命令即可
Restore controlfile from autobackup
Restore controlfile from ‘file name’

5.2 特殊情况下的恢复

在假定丢失了恢复目录与控制文件，只剩下备份集与备份片，这个时候，可能只能从文件中恢复了。以下是调用dbms_backup_restore包，从文件中恢复的例子。

```
declare
devtype varchar2(100);
done boolean;
recid number;
stamp number;
fullname varchar2(80);
begin
devtype :=
dbms_backup_restore.deviceallocate('sbt_tape',params=>'ENV=
(NSR_SERVER=backup_server)');
dbms_backup_restore.restoresetdata file;
dbms_backup_restore.restorecontrolfileto(
'first_control_file');
dbms_backup_restore.restorebackuppiece('backup_piece', done);
dbms_backup_restore.copycontrolfile ('first_control_file',
'second_control_file', recid, stamp,fullname);
-- repeat the above copycontrolfile for each control file
end; /
```

5.3 还原检查与恢复测试

与备份检查一样，还原操作也可以检查是否能正常 restore 或者是否该备份集是否有效。如：
RMAN> RESTORE DATABASE VALIDATE;
RMAN> VALIDATE BACKUPSET 218;
Recover 还可以进行测试，检测恢复的错误，错误信息记载在 alert 文件中，通过测试，我们可以知道该恢复操作是否能正常完成。

SQL>RECOVER TABLESPACE sales TEST;
SQL>RECOVER DATABASE UNTIL CANCEL TEST

5.4 块级别的恢复

块恢复 Block Media Recovery (BMR)，块是恢复的最小单元，通过块可以减少恢复时间，而且数据文件可以在线。恢复块的时候，必须指定具体的块号，如：
BLOCKRECOVER datafile 6 BLOCK 3;

要恢复的坏块信息可以从报警与跟踪文件，表与索引的分析，DBV 工具或第三方媒体管理工具以及具体的查询语句中获得。产生块损坏的原因一般是间断或随机的 IO 错误或者是内存的块错误。

块的错误信息保存在 V$DATABASE_BLOCK_CORRUPTION，用如下命令恢复该视图中列出的坏块

```
RMAN> BLOCKRECOVER CORRUPTION LIST
2> RESTORE UNTIL TIME 'sysdate – 10';
```

备份的坏块信息保存在
V$BACKUP_CORRUPTION
V$COPY_CORRUPTION

可以用如下的命令来恢复坏块。

```
BLOCKRECOVER datafile 2 BLOCK 12, 13 datafile 7 BLOCK 5, 98, 99 datafile 9 BLOCK 19;
BLOCKRECOVER TABLESPACE SYSTEM DBA 4194404, 4194405 FROM TAG "weekly_backup";
BLOCKRECOVER TABLESPACE SYSTEM DBA 4194404, 4194405 RESTORE UNTIL TIME 'SYSDATE-2';
```

5.5 数据库复制

可以用 RMAN 来进行数据库的复制与克隆，RMAN 提供一个专门的命令来完成这个操作。如

```
CONNECT TARGET
CONNECT AUXILIARY SYS/aux_pwd@newdb
DUPLICATE TARGET DATABASE TO ndbnewh
  LOGFILE
    '?/dbs/log_1.f' SIZE 100M,
    '?/dbs/log_2.f' SIZE 100M
  SKIP READONLY
  NOFILENAMECHECK;
```

在以上的命令执行之前，注意如下几点

1、备份主库上的所有数据文件，控制文件以及备份时与备份后产生的归档日志，并把该备份拷贝到需要复制的机器同样的目录下（如果不是同样的目录，在 linux/unix 环境下可以考虑建立一个链接来完成）。

2、拷贝主数据库的初始化参数文件到复制的机器上，并做相应的修改，如修改数据库名称与实例名称

3、在要复制的机器上创建新的密码文件，并启动复制的数据库到 nomount 下。
4、配置主数据库到复制数据库的网络连接或者复制数据库到主数据库的连接。
5、在主数据库或者复制的数据库上运行 RMAN，分别连接主数据库与复制数据库实例。
6、运行复制命令，命令将还原所有数据文件，重新创建控制文件，并利用新的参数文件启动恢复数据库到一致状态，最后用 resetlog 方式打开数据库，创建指定的 redolog。
复制命令也可以从磁带上的备份进行复制，并改变数据库名称，也可以改变数据库文件的新的路径以及恢复到以前的时间点，跳过不需要复制的表空间等，如一个比较复杂的复制命令：

```
RUN
{
   ALLOCATE AUXILIARY CHANNEL newdb1 DEVICE TYPE sbt;
   DUPLICATE TARGET DATABASE TO newdb
   DB_FILE_NAME_CONVERT=('/h1/oracle/dbs/trgt/','/h2/oracle/oradata/newdb/')
   UNTIL TIME 'SYSDATE-1'    # specifies incomplete recovery
   SKIP TABLESPACE cmwlite, drsys, example      # skip desired tablespaces
   PFILE = ?/dbs/initNEWDB.ora
   lOGFILE
         GROUP 1 ('?/oradata/newdb/redo01_1.f',
                  '?/oradata/newdb/redo01_2.f') SIZE 200K,
         GROUP 2 ('?/oradata/newdb/redo02_1.f',
                  '?/oradata/newdb/redo02_2.f') SIZE 200K
         GROUP 3 ('?/oradata/newdb/redo03_1.f',
                  '?/oradata/newdb/redo03_2.f') SIZE 200K REUSE;
}
```

5.6 利用 RMAN 创建备用数据库

利用 RMAN 创建备用数据库可以用两种办法，一种是常规的 Restore 命令，利用从主数据库拷贝过去的备用控制文件，把备用数据库启动到备用 mount 下，这个时候的备用数据库是没有数据文件的。然后在备用端，启动 RMAN 命令，连接该数据库（与主数据库 DBID 一样），把从主数据库拷贝过来的 RMAN 备份还原出来。最后就与其它方法一样了，进入备用的管理恢复模式。
另外一个办法就是复制命令了，如
DUPLICATE TARGET DATABASE FOR STANDBY NOFILENAMECHECK;
以下详细的介绍了这一个过程。
1、创建备用参数文件与密码文件，启动备用数据库到 nomount 下
2、备份主数据库与备用控制文件以及所有归档

```
RMAN> Backup Database;
RMAN> Backup current controlfile for standby;
RMAN> sql "Alter System Archive Log Current";
RMAN> Backup filesperset 10 ArchiveLog all delete input;
```

3、拷贝所有的备份到备用数据库相同路径下
4、配置主数据库到备用数据库的连接
5、启动 RMAN

```
rman target / auxiliary sys/change_on_install@STANDBY
```

6，开始创建备用数据库

RMAN> duplicate target database for standby dorecover nofilenamecheck;

整个过程包括了备用控制文件的创建，启动到 Mount 下，参数文件中指定的路径转换与数据文件的还原，归档日志的还原等。

7、最后恢复日志并启动到管理恢复模式下。

SQL> recover standby database;

SQL> alter database recover managed standby database disconnect;

六、RMAN 的管理

6.1Report 命令

Report 命令可以检测那些文件需要备份，那些备份能被删除以及那些文件能不能获得的信息，如

报告数据库的所有能备份数据文件对象

Report schema

或者

RMAN> REPORT SCHEMA AT TIME 'SYSDATE-14';

RMAN> REPORT SCHEMA AT SCN 1000;

RMAN> REPORT SCHEMA AT SEQUENCE 100 THREAD 1;

报告需要备份的数据文件

Report need backup [redundancy | days | incremental n];

报告过期了的数据文件或者不可用的备份与拷贝

Report obsolete [orphan]

报告不能获得或者不能到达的数据文件信息

Report unrecoverable [database]

6.2 List 命令

List 命令一般用来查看备份与拷贝信息，如

查看备份信息

List backup

查看备份汇总信息

List backup summary

查看文件拷贝的信息

List copy

查看具体的备份信息

List backup of datafile ‘file name’

list incarnation of database;

6.3 Crosscheck 命令

检查磁盘或磁带上的备份或拷贝是否正确，并更新备份或者拷贝的状态
如果不正确，将标记为 expired（过期）
Crosscheck backup;
Crosscheck archivelog all;
Delete [noprompt] expired backup 命令删除过期备份
也可以用 List 来查看相应的报告
LIST EXPIRED BACKUP;
LIST EXPIRED BACKUP SUMMARY;

6.4 Delete 命令

Delete 命令可以用来删除指定的备份或者用来删除废弃或者是过期的备份集
如删除指定的备份集与备份片
RMAN> DELETE BACKUPPIECE 101;
RMAN> DELETE CONTROLFILECOPY '/tmp/control01.ctl';
RMAN> DELETE BACKUP OF TABLESPACE users DEVICE TYPE sbt;
删除过期或者废弃了的备份
RMAN> DELETE EXPIRED BACKUP;
RMAN> DELETE NOPROMPT OBSOLETE;
RMAN> DELETE OBSOLETE REDUNDANCY = 3;
RMAN> DELETE OBSOLETE RECOVERY WINDOW OF 7 DAYS;
删除指定的备份归档
RMAN> DELETE NOPROMPT ARCHIVELOG UNTIL SEQUENCE = 300;

七、恢复目录

Oracle 版本 9 因为控制文件的自动备份，可以很大程度成不需要使用恢复目录，但是使用恢复目录的也有如下好处
- 有些命令只被恢复目录支持（对于 9i 来说，也就是专门操作恢复目录的语句而已）
- 能保留更多的历史备份信息
- 一个恢复目录能管理与备份多个目标数据库
- 如果在 9i 以前，丢失控制文件而没有恢复目录将是难以恢复的
- 如果没有恢复目录，而且发生了结构上的改变，时间点的恢复需要小心操作
- 能存储备份与恢复的脚本

可以看到，主要是可以保留更多的备份信息与方便的管理多个目标数据库，这个在众多目标数据库的情况下，是可以考虑的。

7.1 创建恢复目录

注意，恢复目录不要与目标数据库在同一台机器上，而且大小要求比较小。

SQL> create user RMAN identified by RMAN
2 temporary tablespace TEMP
3 default tablespace RCVCAT
4 quota unlimited on RCVCAT;
SQL> grant recovery_catalog_owner to RMAN;
RMAN> create catalog
RMAN> register database;

恢复目录可以采用如下命令升级与删除

RMAN> UPGRADE CATALOG;
RMAN> DROP CATALOG;

7.2 恢复目录管理

恢复目录支持如下的命令

{CREATE|UPGRADE|DROP} CATALOG
{CREATE|DELETE|REPLACE|PRINT} SCRIPT
LIST INCARNATION
REGISTER DATABASE
REPORT SCHEMA AT TIME
RESET DATABASE
RESYNC CATALOG

1、Resync 命令

Resync 可以同步数据库与恢复目录之间的信息，在实际情况下，rman 一般可以自动同步。

在如下情况下需要同步

- 数据库物理结构的改变
- 数据文件增加或者是改变大小
- 表空间删除
- 回滚段的创建与删除
- 每产生 10 个归档日志

2、Reset 命令

目标数据库 resetlogs 之后，需要重新设置恢复目录。Reset 命令就用来重新设置恢复目录。

7.3 恢复目录视图

恢复目录本身有一组视图，用于存放目标数据库与备份信息，如

恢复目录的相关视图

RC_DATABASE
RC_DATAFILE
RC_STORED_SCRIPT

RC_STORED_SCRIPT_LINE
RC_TABLESPACE
可以通过如下命令来查看相关信息
select * from rc_database;

7.4 存储脚本

存储脚本
```
RMAN> creata script level0backp{
        backup
        incremental level 0
        format '/u01/db01/backup/%U'
        filesperset 5
        database plus archivelog delete input;
        sql 'alter database archive log current';
        }
```
执行脚本
```
RMAN> run {execute script Level0backup;}
```
更新脚本
```
RMAN> replace script level0backup{
        ……
        }
```
删除脚本
```
RMAN> delete script Level0backup;
```
查看脚本
```
RMAN> print script level0backup;
```

小节

RMAN的体系结构与运行方法
RMAN的常见备份方法
RMAN的还原与恢复方法
RMAN的管理方法与常见命令
恢复目录与恢复目录的使用

附录

废弃的命令与取代的新命令

版本	老的命令	当前语法
9.2	REPLICATE	RESTORE CONTROLFILE

		FROM ...
9.2	SET AUTOLOCATE	Now enabled by default.
901	ALLOCATE CHANNEL FOR DELETE	Not available
901	ALLOCATE CHANNEL ...TYPE	CONFIGURE CHANNEL ... DEVICE TYPE
9.0.1	ALLOCATE CHANNEL ...KBYTES	CONFIGURE CHANNEL ...MAXPIECESIZE
901	ALLOCATE CHANNEL ...READRATE	CONFIGURE CHANNEL ... RATE
9.0.1	... ARCHIVELOG ...LOGSEQ	… ARCHIVELOG ... SEQUENCE
9.0.1	BACKUP ... SETSIZE	BACKUP ... MAXSETSIZE
9.0.1	CHANGE ... CROSSCHECK	CROSSCHECK
9.0.1	CHANGE ... DELETE	DELETE
9.0.1	REPORT ... AT LOGSEQ	REPORT ... AT SEQUENCE
9.0.1	SET AUXNAME	CONFIGURE AUXNAME

一个实用脚本，包括备份 RAC 数据库与归档日志的 shell 脚本

```
[oracle@db worksh]$ more rmanback.sh
#!/bin/sh
#set env
export ORACLE_HOME=/opt/oracle/product/9.2
export ORACLE_SID=db2in1
export NLS_LANG="AMERICAN_AMERICA.zhs16gbk"
export PATH=$PATH:$ORACLE_HOME/bin:/sbin:/usr/sbin

echo "-----------------------------start-----------------------------";date
#backup start
$ORACLE_HOME/bin/rman <<EOF
connect target
delete noprompt obsolete;
backup database include current controlfile format '/rmanback/db2/%U_%s.bak' filesperset = 2;

run{
ALLOCATE CHANNEL node_c1 DEVICE TYPE DISK CONNECT 'sys/pass@db1in1';
ALLOCATE CHANNEL node_c2 DEVICE TYPE DISK CONNECT 'sys/pass@db2in2';
sql 'ALTER SYSTEM ARCHIVE LOG CURRENT';
backup archivelog all delete input format '/ rmanback/db2/%U_%s.bak' filesperset = 5;
}

list backup;
exit;
EOF
echo "------------------------------end------------------------------";date
```